

Systembeschreibung

Wir haben fünf verschiedenen Packages wobei jede eine andere Aufgabe hat.

Das GUI Package regelt das erstellte Fenster mit den nötigen Styles mittels einer zentralen Klasse namens `WorkshopControl.java` und den dazugehörigen FXML-Dateien sowie den dazugehörigen CSS-Dateien. Die Klasse hat als Attribute die Objekte des Fensters, sprich Labels, TextAreas usw., sowie eine Liste mit Aufgaben die aus dem Katalog geladen werden mit Hilfe von entsprechenden Klassen auf die hier später eingegangen wird. Unter anderem hat die Klasse noch die Attribute `Timer`, `Phase` und `StatsManager` (auf `StatsManager` wird später eingegangen). Diese Attribute sind für das Regeln der Logik mit den Erweiterungen wichtig.

Diese Klasse `WorkshopControl.java` hat noch zwei innere Klassen, `Timer` und `Phase`. Der Sinn von `Timer` ist es einen die Zeit dem User anzuzeigen und falls `Babysteps` eingeschaltet sind für das regeln der `Babysteps`. Der Sinn von der Klasse `Phase` ist es die derzeitigen Phasen anzugeben, sowie das hin und her wechseln zwischen den einzelnen Phasen.

Beim Erstellen der Liste mit den Auswählbaren Aufgaben kommen die anderen Packages ins Spiel. Hier werden durch das Package `XML` und `util` in diese Liste Objekte der Klasse `Exercise` geladen.

Die Klasse `Exercise.java` in `util` hat alle Informationen von der jeweiligen Aufgabe gespeichert, also Name der Übung, Beschreibung, Klassenvorlagen usw.

Die Klasse `CodeCompiler.java` stellt eine statische Methode zur Verfügung die testet ob ein String kompiliert ohne Fehler und ob die Tests bestanden werden.

Die Reader des Katalogs erstellen diese Objekte von `Exercise`. Also falls eine Aufgabe fehlerhaft im Katalog ist wird diese falsch oder gar nicht als Aufgabe geladen, deshalb sollte auf die Korrektheit der Aufgaben im Katalog geachtet werden.

Das Package `statistics` hat eine Klasse `StatsManager.java`. Diese ist für das Timetracking wichtig und erstellt ein Chart für die bearbeitete Aufgabe in `WorkshopControl.java` indem es die Zeiten in den einzelnen Phasen ausgibt als Chart. Dabei werden alle Berechnungen vor dem User geheim gehalten und nur intern gemacht.

Das letzte Package hat eine Klasse `Main.java` die erstellt wurde um die GUI zu laden und die Stage zu setzen. Im Großen und Ganzen waren das die wichtigsten Klassen die das Programm zum Laufen bringen. Die Logik der gestellten Aufgabe passiert in der `WorkshopControl.java` wobei diese auch das Design regelt welches frei wählbar ist (ob `dark` oder `white`) und die anklickbaren Aufgaben gibt sowie das deaktivieren der Erweiterungen (`Babysteps`, `Timetracking`). Die anderen Klassen sind um die Aufgaben zu laden oder zum testen ob der Code kompiliert oder für das Timetracking.